International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)
ISSN(P): 2249-6890; ISSN(E): 2249-8001
Vol. 7, Issue 1, Feb 2017, 1-12




# MULTI OBJECTIVE FOR A PARTIAL FLEXIBLE OPEN SHOP SCHEDULING PROBLEMS USING HYBRID EVOLUTIONARY ALGORITHM

**N. JANANEESWARI[1], S. JAYAKUMAR[2] & M. NAGAMANI[3]**

[1]*Research Scholar of Mathematics, Aringar Anna government Arts college, Tiruvannamalai, Tamil Nadu, India*

[2]*Assistant Professor of Mathematics, Aringar Anna government Arts college, Tiruvannamalai, Tamil Nadu, India*

[3]*Associate Professor of Mathematics, Global Institute of Engineering & Technology, Vellore, Tamil Nadu, India*

***ABSTRACT***

***A Partial Flexible Open Shop Scheduling Problem (FOSP) is an optimization problem in which ideal jobs are assigned to resources at particular Machine and times. As of late, many attempts have been made in the solution of this issue utilizing a different scope of instruments and methods. This paper presents a hybrid evolutionary algorithm (HEA) for FOSP. The hybrid algorithm is a combination between evolutionary algorithm (EA) and local search. Firstly, another initialization technique is proposed. Also in view crossover and mutation operators utilized. Secondly, local search based on the neighbourhood structure is applied in the EA result. At long last, the approach is tried on an arrangement of standard occasion taken from the writing. The calculation comes about have approved the viability of the proposed approach.***

***KEYWORDS: Multi Objective, Partial Flexible, Evolutionary Algorithm, Open Shop Scheduling Problem & Local Search***

**Received:** Dec 08, 2016; **Accepted:** Jan 11, 2017; **Published:** Jan 12, 2017; **Paper Id** IJMPERDFEB20171

Original Article

## INTRODUCTION

Scheduling has turned into a basic factor in many open shops, particularly for true modern applications [1-4]. Discovering a new scheduling to achieve the work in a minimum time and all the most proficiently is called FOSP. The FOSP can be depicted in the accompanying the most proficiently arrangement of jobs and a set of machines. Each accompanying the, one job at once. Every job is comprised of a chain operations, each of which should be handled during an uninterrupted continuous days of a given length on a given particular machine. The intention is to discover a schedule, that is, an assignment of the operations to time intervals on the machines that has a term required to minimum complete [5]. The FOSP is among the hardest combinatorial problems. In addition to the fact that is confused, but it is one of the hardest and exceedingly awful NP-complete individuals. As a rule, scheduling problems are NP; NP stands for non-deterministic polynomial, which means that it is not possible to solve an arbitrary instance in polynomial time. So, the FOSP has garnered attention due to both its practical importance and its solution complexity [6, 7].

At present, the method for the FOSP mainly includes two kinds, one of which is exact methods and the other approximation methods. Exact techniques are branch and bound, decomposition methods and linear programming have been fruitful in solving little cases [8].Manufacturing framework, most scheduling issues are exceptionally perplexing in nature and extremely confused to be settled by correct techniques it turns out to be monotonically increasingly essential to investigate most ideal methods for getting better schedules that incorporate

priority dispatch, moving bottleneck approach, meta- heuristic strategies and local search. Starting late, various abnormal techniques have been utilized to control different heuristics, known as meta-heuristics prompted to better and more refreshing outcomes [9 - 17]. Therefore, a number of meta-heuristics were proposed in literature for FOSP such as EA [18, 19] simulated annealing (SA) [20, 21] ant colony optimization (ACO) [22-24] tabu search (TS) [25, 26], particle swarm optimization (PSO) [27, 28] and Consultant Guided Search algorithm (CGS) [29].

One of Meta heuristic methods is the EA. EA propelled by the procedure of Darwinian evolution, has been perceived as a general enquiry system and an improved technique which is regularly valuable for attacking combinational issues. Rather than other Meta heuristic strategies the EA uses a population of arrangements in its search that is difficult to fall in neigh bour hood minima. EA has been used with increasing frequency to address scheduling problems. In [30] Lazar introduced a review of frequent approaches and methods for FOSP which most commonly are used in solving this problem. From this review, we can say that EA is an effective meta heuristic to take care of combinatorial best optimization problems, and has been effectively received to illuminate the FOSP. How to adapt EAs to the FOSPs is very challenging but frustrating. Many efforts have been made in order to give an efficient implementation of EAs to the problem. In [31], a new EA is presented to solve the FOSP, while in [(5)] the impact of random initialization of solving the FOSP is addressed and using EA as an optimization technique.

Due to the NP-hard nature of the FOSP, using simple EA to solve the difficult problem may not be more efficient in practice. Much effort in the literature has focused on hybrid methods [32-37]. Ren and Yupingin [32] design some EA operators (mixed selection operator, new crossover operator and a mutation operator based on the critical path) and solve FOSP more effectively. In [33] Athanasius and Stavros proposed a new hybrid parallel EA with particular hybrid mutation administrators using way-relining ideas from combinatorial optimization approaches. Some researchers focused their attention on combining the EA with local search schemes to develop some hybrid optimization strategies for FOSP [37, 38]. In this paper, we propose an effective hybrid algorithm for FOSP based on EA and local search. Firstly, we design an EA model for FOSP. We propose a new initialization technique, modified crossover and mutation operators to improve the solution quality. Then a local search based on the neighbourhood structure is applied in the EA result [36]. The described approach is tried by an arrangement of standard example taken from the drawing from the OR-library [38, 39]. The rest of the paper is sorted out as takes after. Detailed description of FOSP is presented. EA is briefly introduced. The proposed algorithm is presented is discusses the numerical results. Finally, we summarize the paper and present our future work.

The FOSP can be formulated as follows for "n" occupations, each one is made out of a few operations that must be executed on "m" machines. For every operation utilizes just a single machine for a fixed duration. Every machine can handle at most one operation at once and once an operation begins preparing on a given machine, it must finish handling on that same machine with no intrusion. The operations of a given occupation must be handled in a given arrangement of request [5]. The issue's fundamental point is to plan the exercises or operations of the occupations on the machines with the end goal that the aggregate time to complete all employments, that is the make traverse (Cmax), is minimized. The term makes span alludes to the aggregate time to finish every one of the operations of all employments on all machines. It is a measure of the day and age from the beginning time of the main operation to the consummation time of the last operation. In some cases there might be various arrangements that have the base make traverse, yet the objective is to discover any of them: it is not important to locate all conceivable optimal solutions.

## PROBLEM FORMULATION

Let $J = \{0,1..., n\}$, denote the set of operations to be scheduled and $M = \{1,..., m\}$ the set of machines. The operations 0 and $n$ 1are dummy have no duration and represent the initial and final operations. The operations are interrelated by two kinds of constraints. First, the precedence constraints, which force each operation to be scheduled after all predecessor operations, $Q_j$ are completed. Second, operation can only be scheduled if the machine it requires is idle. Further, let $e_j$ denote the (fixed) duration (processing time) of operation $j$. Let $G_j$ represent the finish time of operation $j$. A schedule can be represented by a vector of finish time ($G_1$, $G_2$,..., $G_{n\ 1}$ ). Let $B$ (t) be the set of operations being processed at time $t$, and let if $s_{j,m}$ 1 operation $j$ requires machine $m$ to be processed and $s_{j,m}$ 0 otherwise [(9)].The conceptual model of the FOSP can be described the following way:

$$\text{Minimize } G_{n+1}(\text{Cmax}) \tag{1}$$

$$G_K \le G_j - e_j \quad j = 1,2,....., n+1 \qquad k \in p_j \tag{2}$$

$$\sum_{j \in B(t)} s_{j,m} \le 1 \quad m \in M \qquad t \ge 0 \tag{3}$$

$$G_j \ge 0 \qquad j = 1,2,.....,n+1. \tag{4}$$

The goal work (1) minimizes the complete time of operation n+1, and along these lines minimizes the make span. Limitations (2) force the priority relations amongst operations and imperative (3) express the one machine can just process one operation at once. At long last, (4) constraints the complete circumstance to be non negative. The FOSP can be formally depicted by a disjunctive graph G B; C D, Where V is an arrangement of nodes speaking to operations of the jobs together with two extraordinary nodes, a source (0) and a sink (*), speaking to the start and end of the schedule, individually. C is an arrangement of conjunctive curves speaking to technological succession of the operations. D is an arrangement of disjunctive curves speaking to set of operations that must be performed on the similar machines. As example, Tables 1, 2 present the data for FOSP with three jobs and three machines. Table 1 includes the routing of each job through each machine while Table 2includes the processing time for each operation presents the disjunctive graph of this example [39, 40].

**Table 1: Machine Sequence**

| Jobs | Machines | | |
|---|---|---|---|
| $J_1$ | $M_1$ | $M_2$ | $M_3$ |
| $J_2$ | $M_1$ | $M_3$ | $M_2$ |
| $J_3$ | $M_2$ | $M_1$ | $M_3$ |

**Table 2: Processing Time**

| Jobs | Machines | | |
|---|---|---|---|
| $J_1$ | 3 | 3 | 3 |
| $J_2$ | 2 | 3 | 4 |
| $J_3$ | 3 | 2 | 1 |

## PROPOSED WORK

### Evolutionary Algorithm

EA is a capable of pursuit procedure in light of regular natural biological evolution which is utilized for finding an ideal or closed ideal solution. The idea of EA was first proposed by Holland [(41)] in the mid 1970s and from the point forward has been generally utilized in solving optimization problem. EA has been implemented successfully in many scheduling problems. EA begins with a arrangement of potential solutions (best chromosomes). Next, genetic search administrator for ex, selection, crossover and mutation are then connected consistently to get another era of chromosomes in which the normal quality over every one of the chromosomes is superior to that of underlying era [42]. This procedure is rehashed until the end standard is met, and the better chromosome of the last era is accounted for as the last solution.

Then the evolutionary search steps can be described as follows [43]

### The Proposed HEA Algorithm

The planned algorithm is a grouping amongst EA and local search. Another method for instating the number of population of arrangements is outlined. A changed crossover and mutation administrator utilized to enhance the resolution quality. At the point nearby hunt in view of the area structure planned by Nowicki and Smutnicki is connected EA result [44]. At long last, the approach is tried on an arrangement of standard example taken from the writing. The essential thought of the proposed algorithm can be depicted as follows:

### Algorithm for EA

**Step 1:** Create initial population

**Step 2:** Check whether the chromosome is constraint or out of constraint

**Step 3:** Evaluate objective function

**Step 4:** Do

**Step 5:** Generate new population

**Step 6:** Select parents from population and recombine

**Step 7:** Cross over Mutation operators

**Step 8:** Assess new population in target work

**Step 9:** Build new era from best of old population and new population

**Step 10:** While attractive arrangement has been found

### Advanced Evolutionary Algorithm

### Initial Population

For FOSP, the early solution acting a serious role in deciding the excellence of definite solution. Be that as it may, the early population has been delivered arbitrarily, it is not effective. It will need longer look for time to get an optimal solution and also decreases the look for likelihood for an optimal solution giving infeasible solutions [(45)]. In this article, another method for instating primary population of solutions is composed. This new technique give possible solutions and

is exceptionally helpful in extensive issue. We create the basic population (job sequence in machines) by relying upon machine sequence for jobs. Arrange the jobs in machines that have previous succession in this machine. For example, problem in table 1, for machine1 there are two jobs (1, 2) should start in it according to their machine sequence. In our approach we consider. Make the begin in machine1with one of these jobs. At that point finish jobs succession in machine 1 with a similar way. Outlines introduction strategy with disjunctive graph. As appeared, we arrange jobs in machine with the closest jobs from begin point (O). For example, the nearest jobs on machine 1 from start point (O) are job 1 and job 2 (yellow circles). So the job sequence in machine 1 will be (1, 2, 3), or (2, 1, 3). A Similar way is utilized for alternate machines.

**Evaluation**

In FOSP, makespan speaks to a decent extent measure. Makespan of the job (Cmax) is the fulfillment time of the considerable number of jobs where, the timetable with the negligible makespan frequently suggests a high usage

In FOSP, makespan represents a good performance measure. Makespan of the jobs (Cmax) is the completion time of all the jobs where, the schedule with the minimal makespan often implies a high utilization of machines. The makespan objective is selected for comparison and assessing the performance of the generated solutions by EA.

**Create New Population**

Making another population from the present population by positioning the solutions as indicated by the fitness function (makespan). At that point we connected the three operators (selection, crossover, and mutation).

**Ranking**

Ranking individuals according to their fitness value, and returns a column vector containing the corresponding individual fitness value, in order later to establish the probabilities of survival that are necessary for the selection process.

**Selection**

For FOSP as any optimization issue there are few methods for selection. The usually utilized strategies for choice of individuals are roulette wheel selection, rank selection, steady state selection, stochastic universal sampling, so on. Here we utilized the roulette wheel selection [46]. In this technique the parents are chosen based on their fitness. Better chromosomes, are having more chances to be selected as parents. Every individual is assigned a slice of the circular Roulette wheel, and the extent of the slice is proportional to the individual fitness of chromosomes, that is, greater the value, larger the size of slice is. The Roulette wheel algorithm is started by judgment the sum of all chromo some's fitness in the population. Then generate random number from the given population interval. At long last, experience the entire population and sum the fitness. At that point when this sum is more than a wellness criteria value, stop and return this chromosome. Demonstrate Roulette wheel for six individuals having distinctive fitness values. The sixth individual has a higher fitness than some other, it is normal that the Roulette wheel selection will select the sixth individual more than some other individual [42].

**Crossover**

The crossover works on two chromosomes at a once and produces offspring by combining features of both chromosomes. By and large, the execution of the EAs depends, to a great extent, on the act of the crossover operator used. Amid the previous past two decades, a variety of crossover operators have been planned for literal permutation encodings

for FOSP, such as partial-mapped crossover (PMX), order crossover (OX), cycle crossover (CX), position-based crossover, order-based crossover, etc.[(47)]. In this document, we use a modified OX that proposed by Gao and et al. [(48)], where they used OX in operation sequence; here we used it in job sequence for every machine. The modified OX can be depicted as takes after:

**Step 1:** Choose a subsection of job sequence for a machine from one parent at random.

**Step 2: Produce** offspring by replicating the substring of job sequence for a machine into the corresponding positions.

**Step 3:** Delete the jobs that are already in the substring from the second parent. The resulted sequence of jobs contains jobs that the offspring needs.

**Step 4:** Place the jobs into the unfixed positions of the offspring from left to right according to the order of the sequence in the Second parent. The modified OX for one machine.

**Mutation**

Mutation is quite recently used to create little perturbations on chromosomes keeping in the mind the end goal to keep the diversity of population. Amid the most recent decade, there are a few transformation mutation operators have been described for ex such as inversion, insertion, shift mutation, reciprocal exchange and displacement, [48]. Reversal mutation chooses two positions inside a chromosome aimlessly and after that alters the substring connecting these two positions. In this paper we propelled inversion mutation; by picking at least two positions in the jobs sequence for one machine and upset them as appeared. These means are applied to different machines, to get another Offspring.

**New Generation**

In this progression, the new era made by the movement starting population with and progeny population. At that point we take the best people of introductory population with era gap percentage and progeny population.

**Termination Test**

The algorithm is completed either the most extreme number of eras is accomplished, or when the people of the population join, merging happens when every individual position in the population is indistinguishable. In this situation, the crossover will have no further impact. Something else, come back to step 4.3.

**Local Search Procedure**

Utilizing EAs for the FOSP is more often than not with a moderate joining rate and simply to trap into local optimal solutions. Keeping in mind the end goal to upgrade the union speed, we join the EA with neighbourhood look plans to build up some hybrid optimization methodologies for FOSP enhancing the arrangement quality. We utilize the approach of Nowicki and Smutnicki [37, 38].The nearby pursuit method begins by finding the basic way for the arrangement acquired by EA. Basic way can be resolved in the Gantt-Chart representation of the arrangement. The basic way breaks down by various squares where a piece a maximal grouping of adjoining operations that require a similar machine is.

Any operation on the basic way is known as a basic operation. Also, an area is characterized as exchanges of any two successive operations. Exchanging the last two or the initial two basic operations gives best arrangement.

The neighborhood seek strategy is appeared, while this demonstrates the stream graph of the proposed algorithm (HEA).

## NUMERICAL RESULTS

To demonstrate the adequacy of the proposed calculation, it is tried on a set of standard **occasion** (FOSP test problems) taken from the writing [(37), (38 s Table 3. The algorithm is coded in MATLAB 7.8 through on an Intel core (TM) i7-4510u CPU, 2.00GHZ -2.60 GHz processor.

**Table 3: Summary of the Parameters used in the Proposed Algorithm**

| **1.** | **Generation gap** | **0.9** |
|---|---|---|
| 2. | Crossover rate | 0.9 |
| 3. | Mutation rate | 0.6 |
| 4. | Selection operator | Roulette wheel selection single point |
| 5. | Crossover operator | Order crossover |
| 6. | Mutation operator | Inversion mutation |
| 7. | EA generation | 50-1000 |

Summary of the parameters used in the proposed algorithm Table 4 shows the problem name and size(number of jobs ×number of operations), the best known solution (BKS), results obtained by applying EA only, the results obtained by HEA and the improvement percentage in the obtained results due to hybrid local search with EA. In addition, showed the improvement percentage for each problem

The Gantt graphs of some acquired outcomes by our approach for the issues (FT06-LA01-LA05) are represented. From the table and figures, we can state that hybridizing the nearby inquiry with EA enhances the arrangements quality. Where the mean change rate is around 5.36%. Moreover, we can see that the solutions acquired by HEA are better and meet to the BKS than solutions got by EA as it were.

**Table 4: Comparison between BKS and the Proposed Algorithm Results (EA Only and HEA) with Improvement Percent**

| **Problem** | **Size** | **Best Known Solution(BKS)** | **Results of EA** | **Results of HEA** | **Improvement Percent** |
|---|---|---|---|---|---|
| FT06 | 6×6 | 55 | 55 | 55 | 0.000 |
| LA01 | 10×5 | 666 | 666 | 666 | 11.603 |
| LA02 | 10×5 | 655 | 790 | 714 | 11.055 |
| LA03 | 10×5 | 597 | 683 | 617 | 6.779 |
| LA04 | 10×5 | 590 | 672 | 632 | 0.000 |
| LA05 | 10×5 | 593 | 593 | 593 | 0.000 |
| LA06 | 15×5 | 926 | 926 | 926 | 1.910 |
| LA07 | 15×5 | 890 | 916 | 899 | 0.000 |
| LA08 | 15×5 | 863 | 863 | 863 | 0.000 |
| LA09 | 15×5 | 951 | 951 | 951 | 0.000 |
| LA10 | 15×5 | 958 | 958 | 958 | 0.000 |
| LA11 | 20×5 | 1222 | 1222 | 1222 | 0.000 |
| LA12 | 20×5 | 1039 | 1039 | 1039 | 0.000 |
| LA13 | 20×5 | 1150 | 1150 | 1150 | 0.000 |
| LA14 | 20×5 | 1292 | 1292 | 1292 | 0.000 |
| LA15 | 20×5 | 1207 | 1324 | 1311 | 1.077 |
| LA16 | 10×10 | 945 | 1140 | 1077 | 6.667 |
| LA17 | 10×10 | 784 | 848 | 843 | 0.638 |
| LA18 | 10×10 | 848 | 948 | 894 | 6.638 |
| LA19 | 10×10 | 842 | 961 | 896 | 7.719 |
| LA20 | 10×10 | 902 | 956 | 949 | 0.776 |

Summarized in Table 3 The algorithm is coded in MATLAB 7.8 and the simulations have been executed on an Intel core (TM) i7-4510u cpu, 2.00GHZ -2.60 GHz processor. The non-deterministic nature of our algorithm makes it necessary to carry out multiple runs on the same problem instance in order to obtain meaningful results. We ran our algorithm 15times for each test problem. Worst, best, mean and standard deviation (Std.) of results are illustrated in Table 5. The standard deviation is a very important problem in practical production area. In order to ensure our scheduling plan performs effectively, the standard deviation should be as small as possible. Through simulation, we find the standard deviation is small which indicates that HEA possesses excellent robustness.

**Table 5: Mean, Worst Best and Standard Deviation for the Test Instances**

| Problem | Worst | Best | Mean | Standard Deviation (%) |
|---|---|---|---|---|
| FT06 | 55 | 55 | 55 | 0 |
| LA01 | 680 | 666 | 674.30 | 5.021 |
| LA02 | 758 | 714 | 729 | 18.745 |
| LA03 | 671 | 617 | 653.10 | 11.588 |
| LA04 | 672 | 632 | 630.70 | 3.900 |
| LA05 | 593 | 593 | 593 | 0 |
| LA06 | 593 | 926 | 593 | 0 |
| LA07 | 973 | 899 | 934.57 | 22.756 |
| LA08 | 863 | 863 | 863 | 0 |
| LA09 | 951 | 951 | 951 | 0 |
| LA10 | 958 | 958 | 958 | 0 |
| LA11 | 1222 | 1222 | 1222 | 0 |
| LA12 | 1039 | 1039 | 1039 | 0 |
| LA13 | 1150 | 1150 | 1150 | 0 |
| LA14 | 1292 | 1292 | 1292 | 0 |
| LA15 | 1379 | 1311 | 1349.40 | 22.937 |
| LA16 | 1107 | 1077 | 1088.60 | 11.242 |
| LA17 | 877 | 843 | 851.92 | 3.936 |
| LA18 | 954 | 894 | 930.07 | 12.807 |
| LA19 | 1015 | 896 | 961 | 42.007 |
| LA20 | 1056 | 949 | 991 | 18.3011 |

For comparison purposes, Table 6 shows the results and the percentage relative error with respect to BKS obtained by proposed algorithm as compared to CGS [(28)]. In addition, proposed the percentage relative error for the two algorithms CGS and HEA. It is quite evident that our algorithm more converges to the BKS and gives comparable minimum relative error or better than those obtained by CGS; where HEA reached to the BKS for 52.4% of its solved test problems while the CGS reached to the BKS for 26.3% and in the major remained problems HEA reached near to BKS than CGS.

**Table 6: Results and the Percentage Relative Error for the Two Algorithms CGS and HEA**

| Problem | CGS | HEA | Percentage Relation Error [CGS] (%) | Percentage Relation Error [CGA] (%) |
|---|---|---|---|---|
| FT06 | - | 55 | - | 0.000 |
| LA01 | 713 | 666 | 7.057 | 0.000 |
| LA02 | 757 | 714 | 15.572 | 9.008 |
| LA03 | 682 | 617 | 14.238 | 3.350 |
| LA04 | 669 | 632 | 13.389 | 7.119 |
| LA05 | 593 | 593 | 0.000 | 0.000 |
| LA06 | 926 | 926 | 0.000 | 0.000 |
| LA07 | - | 899 | - | 1.685 |
| LA08 | 897 | 863 | 3.939 | 0.000 |
| LA09 | 956 | 951 | 0.526 | 0.000 |
| LA10 | 958 | 958 | 0.000 | 0.000 |
| LA11 | 1222 | 1222 | 0.000 | 0.000 |
| LA12 | 1058 | 1039 | 1.829 | 0.000 |
| LA13 | 1152 | 1150 | 0.174 | 0.000 |
| LA14 | 1292 | 1292 | 0.000 | 0.000 |
| LA15 | 1310 | 1311 | 8.536 | 8.616 |
| LA16 | 1010 | 1077 | 6.878 | 13.968 |
| LA17 | - | 843 | - | 7.526 |
| LA18 | 914 | 894 | 7.783 | 5.425 |
| LA19 | 944 | 896 | 12.114 | 6.413 |
| LA20 | 949 | 949 | 5.211 | 7.761 |

Overall, we applied the proposed algorithm in various test instances with different size. In general, the results have shown the correctness, feasibility and the usability of the proposed algorithm. Noting also that increasing the problem size doesn't guarantee that the proposed algorithm reach to the BKS due to using 2.00GHZ 2.60 GHz processor. But, HEA integrates the powerful global searching of EA with the powerful local searching of local search. In addition, Integrating EA with local search accelerates the optimum seeking operation and speeds the convergence to the BKS. Finally, due to simplicity of HEA procedures, it can be used to handle engineering applications.

## CONCLUSIONS

In this paper we show a combination among EA and local search for tackling the Job-shop scheduling issue. At first, a new introduction technique is proposed. An altered crossover and mutation administrator are utilized. Besides, nearby inquiry in view of the area structure is connected in the EA result. At last, the approach is tried on standard instances taken from the writing. The outcomes acquired by HEA are better and focalize to the BKS than arrangements got by EA just, which imply that hybridizing nearby purist with EA, enhance the solutions quality. Furthermore, the proposed approach is more joins to the BKS and gives practically identical least relative error or better than to those got by other approach. As a rule, the outcomes have demonstrated the accuracy, feasibility and the ease use of the proposed algorithm.

In our future works, accompanying will be investigated for solving bigger scale cases to exhibit the effectiveness of the presented approach. Hybridizing more advance optimization methods with the proposed algorithm to quicken the union property and enhance the arrangement quality. Redesigning the proposed algorithm to explain the FOSP as multi-objective optimization issue.

## ACKNOWLEDGEMENTS

We would like to express or sincere gratitude to the anonymous reviewers for their efforts and comments, which helped us considerably in improving the quality of this paper.

***REFERENCES***

1. *Kolharkar S., ZanwarD.R., (2013). Scheduling in Job Shop Process Industry. Journal of Mechanical and Civil Engineering, 5, 2278-1684.*
2. *PinedoM.L., (2009). Planning and scheduling in manufacturing and services. Springer, second edition, 207-339.*
3. *Chryssolouris G., Subramaniam V.(2001). Dynamic scheduling of manufacturing job shops using genetic algorithms. Journal of Intelligent Manufacturing, 12,281-293.*
4. *Tamilarasi A., Jayasankari S. (2012). Evaluation on GA based Model for solving JSSP. International Journal of Computer Applications, 43(7), 975 – 8887.*
5. *Garey, M.R., Johnson, D.S. and Sethi R., (1976). The Complexity of Flow-shop and Job-shop Scheduling. Mathematics of Operations Research, 1, 117-129.*
6. *Garey M.R., Johnson D.S. (1979). Computers and intractability: A guide to the theory of NP-completeness. First edition, W. H. Freeman & Co. New York, NY, USA, 359-376.*
7. *Goncalves J.F., Mendes J.J.M., Resende M.G.C. (2005). A hybrid genetic algorithm for the job shops scheduling problem. European Journal of Operational Research, 167, 77–95.*
8. *El-Shorbagy M.A., Mousa A.A., Nasr S.M. (2016). A Chaos-Based Evolutionary Algorithm for General Nonlinear Programming Problems. Chaos, Solitons and Fractals, 85, 8–21.*
9. *Farag M.A., El-Shorbagy M.A., El-Desoky I.M., El-Sawy A.A. and Mousa A.A. (2015). Binary-Real Coded Genetic Algorithm Based K-Means Clustering for Unit Commitment Problem. Applied Mathematics, 6, 1873-1890.*
10. *Mousa A.A., El_Desoky I.M. (2013). Stability of Pareto Optimal Allocation of Land Reclamation by Multistage Decision-Based Multi pheromone Ant Colony Optimization. Swarm and Evolutionary Computation, 13, 13–21.*
11. *Mousa A.A., El-Shorbagy M.A., Abd El-WahedW.F. (2012). Local Search Based Hybrid Particle Swarm Optimization for Multiobjective Optimization. International journal of Swarm and evolutionary computation, 3, 1-14.*
12. *Mousa A.A., El-Shorbagy M.A. (2012). Enhanced Particle Swarm Optimization Based Local Search for Reactive Power Compensation Problem. Applied Mathematics 3, 1276-1284.*
13. *El-Sawy A.A., Hendawy Z.M., El-Shorbagy M.A.(2012).Trust-Region Algorithm Based Local Search for Multi-Objective Optimization. Proceedings of the first International Conference on Innovative Engineering Systems (ICIES), Egypt, 207-212.*
14. *Abd El-Wahed W.F., Mousa A.A., El-Shorbagy M.A. (2011). Integrating Particle Swarm Optimization with Genetic Algorithms for Solving Nonlinear Optimization Problems. Journal of Computational and Applied Mathematics, 235, 1446–1453.*
15. *El-Shorbagy M.A., Mousa A.A., Abd-El-Wahed W.F. (2011). Hybrid Particle Swarm Optimization Algorithm for Multi-Objective Optimization. Lambert academic publishing GmbH &Co.kG, Berlin.*
16. *Osman M.S., Abo-Sinna M.A., Mousa A.A. (2006). IT-CEMOP: An Iterative Co-Evolutionary Algorithm for Multiobjective Optimization Problem with Nonlinear Constraints. Journal of Applied Mathematics & Computation (AMC), 183, 373-389.*

17. *Zhang C., Rao Y., Li P. (2008). An effective hybrid genetic algorithm for the job shop scheduling problem. The International Journal of Advanced Manufacturing Technology, 39, 965–974.*
18. *Yan-Fang Y., Yue Y. (2015). An improved genetic algorithm to the job shop scheduling problem. Journal of Chemical and Pharmaceutical Research, 7(4), 322-325.*
19. *Song S.-Z., Ren J.-J., Fan J.-X. (2012). Improved simulated annealing algorithm used for job shop scheduling problems. Advances in Electrical Engineering and Automation, Springer, 17–25.*
20. *Thamilselvan R., Balasubramanie P. (2012). Integrating Genetic Algorithm, Tabu Search and Simulated Annealing for Job ShopScheduling Problem. International Journal of Computer Applications, 48(5), 975 – 888.*
21. *Mehmood N., Umer M., Ahmad R., Rafique A.F. (2013).Optimization of Makespan and Mean Flow Time for Job Shop Scheduling Problem FT06 Using ACO. Life Science Journal, 10(4), 477-484.*
22. *Nazif H. (2015). Solving job shop scheduling problem using an ant colony algorithm. Journal of Asian Scientific Research, 5(5), 261-268.*
23. *FlórezE., Gómez W.,Bautista L. (2013). An ant colony optimization algorithm for job shop scheduling problem. International Journal of Artificial Intelligence & Applications, 4(4), 53-66.*
24. *Jaziri W. (2008). Some New Results on Tabu Search Algorithm Applied to the Job-Shop Scheduling Problem, Tabu Search. I-Tech Education and Publishing.*
25. *Zhang C., Li P., Guan Z., Rao Y. (2007). A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. Computers & Operations Research, 34(11), 3229–3242.*
26. *Sha D.Y., Hsu C.-Y. (2006). A hybrid particle swarm optimization for job shop scheduling problem. Computers & Industrial Engineering, 51, 4791–808.*
27. *Zhang G., Shao X., Li P.,Gao L.(2009).An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering, 56, 1309–1318.*
28. *Deepanandhini D., Amudha T. (2013). Solving job-shop scheduling problem with consultant guided search metaheuristics. International Journal of Software and Web Sciences, 3 (1), 1-6.*
29. *Lazár I. (2012). Review on solving the job shop scheduling problem techniques: recent development and trends. Transfer inovácií, 23, 55-60.*
30. *LI Y., CHEN Y. (2010). A Genetic Algorithm for Job-Shop Scheduling. Journal of software, 5, 269-247.*
31. *Qing-dao-er-jiR., Wang Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. Computers & Operations Research 39, 2291–2299.*
32. *Spanos A.C., Ponis S.T., TatsiopoulosI. P., Christoul.T., Rokou E. (2014). A new hybrid parallel genetic algorithm for the job-shop scheduling problem. International transactions in operational research, 21, 479–499.*
33. *Abu-Srhahn A., Al-Hasan M. (2015). Hybrid Algorithm using Genetic Algorithm and Cuckoo Search Algorithm for Job Shop Scheduling Problem. International Journal of Computer Science Issues, 12, 288-292.*
34. *Moin N.H., Sin O.C., Omar M. (2015). Hybrid Genetic Algorithm with Multiparents Crossover for Job Shop Scheduling Problems. Mathematical Problems in Engineering, 2015.*
35. *OmbukiB.M., Ventresca M. (2004). Local Search Genetic Algorithms for the Job Shop Scheduling Problem. Applied Intelligence, 21, 99-109.*

36. *Camino R.V., Varela R. , González M.A. (2010). Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. Journal of Heuristics, 16, 139-165.*
37. *Wang X., Duan H. (2014).A hybrid biogeography-based optimization algorithm for job shop scheduling problem.Computers & Industrial Engineering, 73, 96–114.*
38. *Beasley E.J. (1990). OR-library: distributing test problems by electronic mail. Journal of the Operational Research Society 41 (11).*
39. *Gen M., Cheng R. (1997). Genetic algorithms and engineering design. Wiley, 1-90.*
40. *Balas E., 1969, Machine sequencing via disjunctive graphs: an implicit enumeration algorithm, Operations Research, 17, 941– 57.*
41. *Rao S.S. (2009). Engineering Optimization: Theory and Practice. Wiley, 3rd edition, 693-702.*
42. *Malhotra R., Singh N., Singh Y. (2011). Genetic algorithms: Concepts, design for optimization of process controllers. Computer and Information Science, 4(2), 39-54.*
43. *Nasr S.M., El-Shorbagy M.A., El-Desoky I.M., HendawyZ.M., Mousa A.A.(2015). Hybrid genetic algorithm for constrained nonlinear optimization problems. British journal of mathematics & computer science, 7(6), 466-480.*
44. *Starkweather T., Whitley D., Mathias K., McDaniel S. (1992). Sequence scheduling with genetic algorithms. Proceedings of the US/German Conference on New Directions for OR in Manufacturing, 130-148.*
45. *Park B. J., Choi H.R., Kim H.S.,(2003). A hybrid genetic algorithm for the job shop scheduling problem. Computers & Industrial Engineering, 45, 597–613.*
46. *Baker J.E. (1987). Reducing bias and inefficiency in the selection algorithm. Proceedings of the second international conference on genetic algorithms. Morgan Kaufmann Publishers, 14-21.*
47. *Cheng R., Gen M., Tsujimura Y. (1999). A tutorial survey of job-shop scheduling problems using genetic algorithms. Computers & Industrial Engineering, 36, 343-364.*
48. *Gao J., Sun L., Gen M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35, 2892 – 2907.*